

# Internet Censorship Circumvention Network Volunteer Relay Module

Edgar A. Cobos

Divide by Zero  
December 10, 2015

### **Abstract**

Freedom of speech and to information are essential for the development and further advancement of the human race. Freedom of speech gives the people power to readily speak up about injustices, corruption or express a dissenting point of views, without the fear of prosecution. On a similar note, freedom to information is what gives people the ability to read and view the broadcast of such injustices, corruption or dissenting point of views. Therefore, freedom of speech and to information provides a type of checks-and-balances between the people and governing body.

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Research</b>	<b>4</b>
<b>3</b>	<b>Overview</b>	<b>5</b>
3.1	Problem Statement . . . . .	5
3.2	Challenges (as an individual) . . . . .	5
3.3	Solutions . . . . .	6
3.4	Your individual work . . . . .	6
3.5	Comparison of your work with your teammates . . . . .	6
3.6	Shortcomings and their reasons . . . . .	6
3.7	Lessons learned . . . . .	7
<b>4</b>	<b>Implementation and results (most important part of report)</b>	<b>8</b>
<b>5</b>	<b>Conclusion and Future Work</b>	<b>9</b>
<b>6</b>	<b>Bibliography/Citation</b>	<b>10</b>
<b>7</b>	<b>Appendix</b>	<b>11</b>

# Section 1

## Introduction

The Internet's ability to connect people and ideas gives it great power. Uniting minds from regions near and far. However, some people in power do not believe that sharing ideas is in their best interest. The fear lies in the result of what the masses will do in light of seeing governmental action that only favors those in power or a way of life that gives power to the people. As a result, there exists several countries with governing bodies that enforce Internet traffic filtering. Replacing search results that would otherwise reflect poorly on the country with those that make them look favorable, or all-together blocking website that show any dissent against the government. Since the government has control over the Internet service providers (ISP), they are able to implement packet filtering at the internet backbone level. This means that there is no way around the connecting gateway. In order to combat such censorship, we must go through the monitored gateway, undetected.

The Tor Project<sup>1</sup> has developed the Tor Anonymous Network aimed at keeping users anonymous on the Internet. Most commonly, Chinese citizens use Tor to circumvent the infamous Great Firewall of China<sup>2</sup>. The Tor Project has created a tool that is invaluable to those who wish to express themselves, but it is not without any flaws. The most notable flaw is the network's inability to properly support video streaming. The issue is caused by the network's structure. Interestingly enough, the same network structure that gives it its anonymous power is also what creates the noted flaw. So what does the inability to stream video content have to do with censorship? It means that users of Tor are limited to the kind of information they can access because Tor is only capable of supporting a subset of all available data.

Tor works in the following way; a request is made, the data is relayed from an entry node to a middle node and lastly to an endpoint node, where the request is fulfilled. The network's issue lies with the amount of relaying needed to remain anonymous. Tor fully relies on other users volunteering as relays. Therefore, they are not able to enforce network requirements in order to become a relay for their network. Due to this fact, the time it takes to relay data from one node to the other is dependent on the underlying volunteers' network. It is highly probable that volunteers are on subpar networks and will therefore add extra latency to the data delivery. We must also remember that there are at least three nodes that must be traversed, adding more latency, before the data packets reach their final destination.

---

<sup>1</sup><https://www.torproject.org/about/overview.html.en>

<sup>2</sup>[https://en.wikipedia.org/wiki/Great\\_Firewall](https://en.wikipedia.org/wiki/Great_Firewall)

Our approach is to implement a network minimizes the number of relays the data has to traverse before reaching its final destination. Our goal is a bit different to that of the Tor Project's, we aim to solely circumvent censoring network by making users anonymous in the eyes of Internet service providers. Our network works in the following way: a request is made, a client application relays the data to a proper volunteer relay then lastly to a static proxy which processes the request. In this approach, as opposed to using three relays with potentially subpar internet speed, we use one. With speed in mind, the volunteer relay module application needs to run as efficiently as possible as to not deter the volunteers or users from the system.

## Section 2

### Research

Include previous closely related work here. Cite them and compare your work in enough details with them. [Very important!]

# Section 3

## Overview

### 3.1 Problem Statement

As part of a circumvention network, the volunteer relay module is an application that will run on volunteering user's computer. The network heavily relies on the volunteer relays to complete a tunnel, for data transfer, from a client to an open internet endpoint. Open internet refers to the type of internet that provides unrestricted access to any available website. Therefore, it is absolutely crucial that the application be simple to use and install. Additionally, that it be disruption free to the volunteer or underlying host operating system.

It should be the case that becoming a relay volunteer is simple and with minimal network setup. Once the relay service is turned on by the volunteer, it should become free of disruptions to the host system. This means that the application will need to run efficiently. On error, it will need to handle the error gracefully. We want to do everything we can to make volunteering favorable. The application should also be able to run on the Linux, Mac and Windows environments.

### 3.2 Challenges (as an individual)

The first challenge was working with the Qt Framework. It was my first time working with this framework, so there were a couple new concepts that I had to learn before being able to properly use it. Though it uses the C++ language for development, it requires that you abide by a specific syntax. Qt uses what they call Slots and Signals to handle function callbacks. I had to manually "connect" a signal to a slot on their respective object. A signal can be emitted by any function call, usually after some work of interest is complete. A slot is called when the signal is emitted and is then tasked with handling the callback. I ran into several issues with this concept initially because the paradigm was completely new to me. Since the Qt framework compiles an application for multiple platforms, I was forced to use wrapper data types and classes. Out of habit, I constantly used the non-wrapper data types and classes, which led to issues.

The application is both a server and client. For the privacy of a users we want to create encrypted links between a client to our application and from our application to the static proxy. This meant that I had to use SSL sockets with the corresponding OpenSSL

library. My original goal was to use the latest OpenSSL library, version 1.0.2e. However, Qt did not make integrating the library easy.

Since the application is going to be handling several connection at once, it would make sense to fork on each new incoming socket connection. The reason for this is that each connection should be independent of the others. If one socket is using up more resources than it should, then it should have no impact on the other connections. Unfortunately, based on Qt's documentation, `fork()` is not available because Windows does not support such operation.

### 3.3 Solutions

Thankfully, Qt is documented<sup>1</sup> extensively and even provides several examples<sup>2</sup> that can be downloaded. Originally, I my goal was to use the latest version of OpenSSL because it provides support for the TLSv1.2 protocol. Qt made it really difficult to incorporate the desired library. Their instructions for including the library involved complex steps to compile the program. It was more involved than time allowed. Therefore, I decided to just stick with the OpenSSL library that comes out of the box with Qt, version 0.9.8. Unfortunately, that version only support the TLS protocol up to version 1.0. However, this protocol is good for now, while the prototype is being developed.

As it turns out, Qt does not support the use of `fork()` because Windows does not have `fork`. In this case, my other option is to use threads. Threads are not the idea solution for our issue since they are all still running under the same process. An issue in one thread may affect the others. In addition to threads, the application makes use of slots and signals to handle connections. Whenever new data is written into the incoming socket buffer, a signal is emitted then a slot reads the buffer data and immediately writes it to the destination socket buffer. The slots and signals ensures that data is being moved from one socket buffer to another to prevent a backup or worse, overwriting.

### 3.4 Your individual work

Highlighting your INDIVIDUAL work is important

### 3.5 Comparison of your work with your teammates

Mention at least 3 issues/weaknesses with your performance and 3 issues/weaknesses with your teammates' performance to be improved in this subsection. A

### 3.6 Shortcomings and their reasons

I have to be able to know exactly what you have done and what you didn't do that you should have done. I expect to see at least a couple of items that you should have done but you couldn't do ([Subsection3.6]). Make sure you state acceptable reasons for not doing those items.

---

<sup>1</sup><http://doc.qt.io/qt-5/gettingstarted.html>

<sup>2</sup><http://doc.qt.io/qt-5/qtexamplesandtutorials.html>



## 3.7 Lessons learned

Afterwards state the lessons you learned in team-work and team projects ([Subsection3.7]). Talk a little about the best and worst parts in your experience in this course project.

## Section 4

# Implementation and results (most important part of report)

Include all of your implementations and results in this section. This section has to contain clear and illustrating figures, charts, etc. No picture or figure can be taken from any sources. Everything has to be your individual work. No figure, charts of your teammates can be included here. At least a few figures here. Follow the guidelines for figures below.

Each picture/figure, sample code and chart can take at most  $1/4$  (25perc) of a page. Having oversized figures, charts, sample codes, etc. severely affects your grade. Make sure your figures are legible and clear yet occupy at most  $1/4$  of a page. If your figures, charts, sample codes are not your own (individual) work you'll lose at least 50

## Section 5

### Conclusion and Future Work

## Section 6

### Bibliography/Citation

## Section 7

## Appendix

Include all of your INDIVIDUAL sample codes and analysis here.